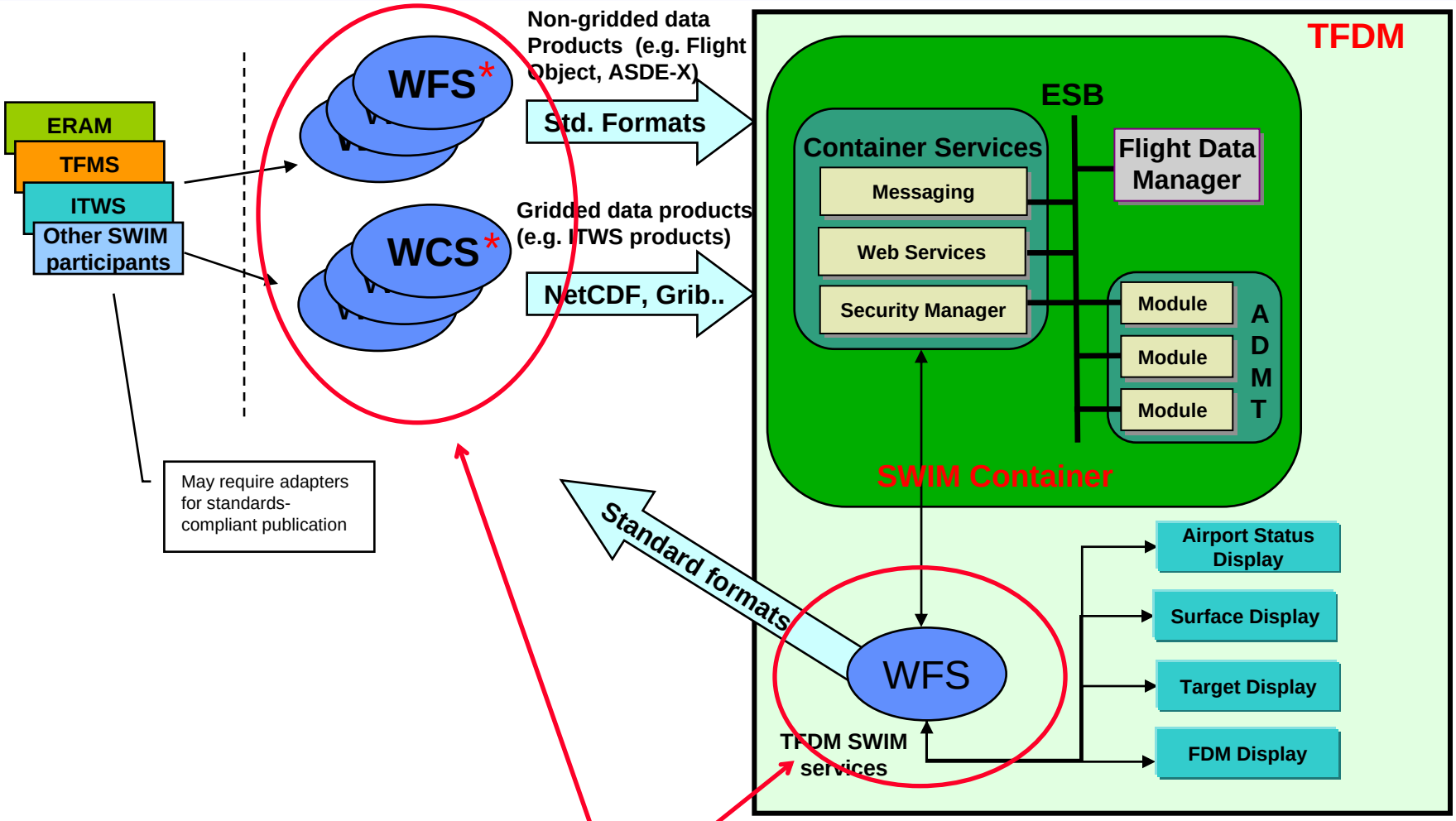
TFDM Information Management

William Moser

MIT Lincoln Laboratory

3 June 2009

SWIM- Integrated Tower Architecture



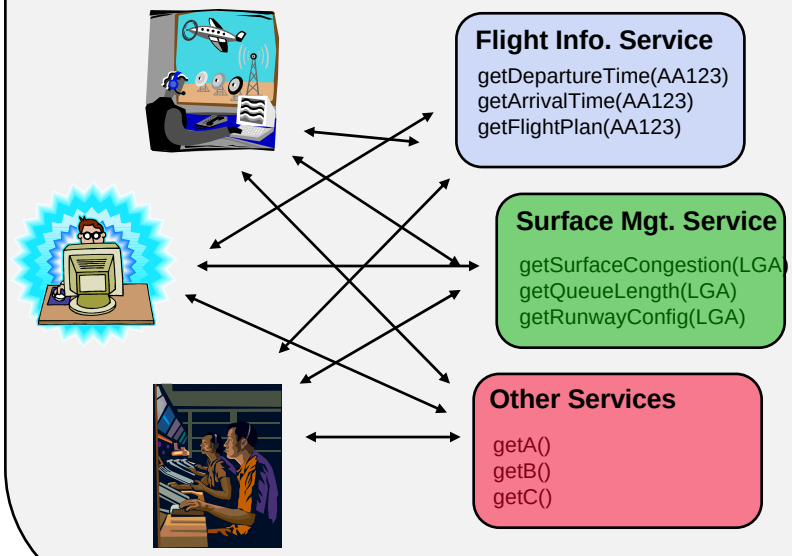
A robust, scalable, and flexible information management architecture is critical to the success of NextGen systems

* Open Geospatial Consortium's Web Feature Service, Web Coverage Service

Scalable Information Architecture

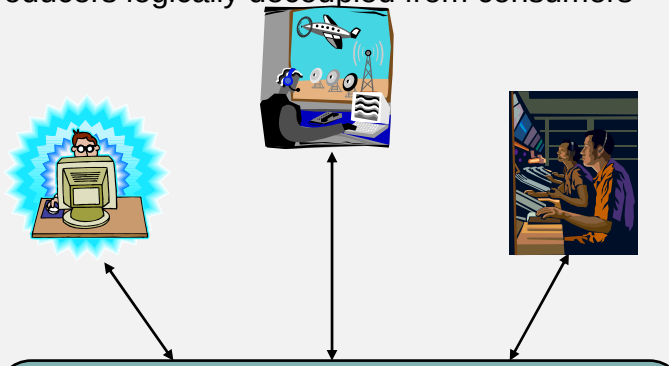
Naïve SOA design is <u>not scalable</u>!

- N-squared problem as more users and service providers connect
- Semantic divergence as service providers provide variations on the same theme (e.g. pub/sub)



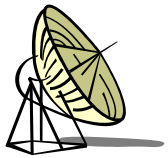
TFDM Information Management Architecture (TIMA) <u>scales well</u> as users and services are added

- Filtered data access interface makes it easy for users to ask for the data they need
- Producers logically decoupled from consumers



Many distinct services **VS** **Single flexible service**

The TIMA Design



A/DMT, ERAM, ATC personnel, etc

Publishers insert data incrementally into XML documents. Insertions are validated against schema definitions.

Consumers receive filtered XML-formatted date via *persistent query* subscription mechanism.

Pub/Sub enabled WFS Server

XML insertions/retrievals mapped to/from database schema using Oracle's XMLDB tools

Design-time tools make it easy to map XML schema to database schema

Data automatically aged out of real-time database, into data warehouse

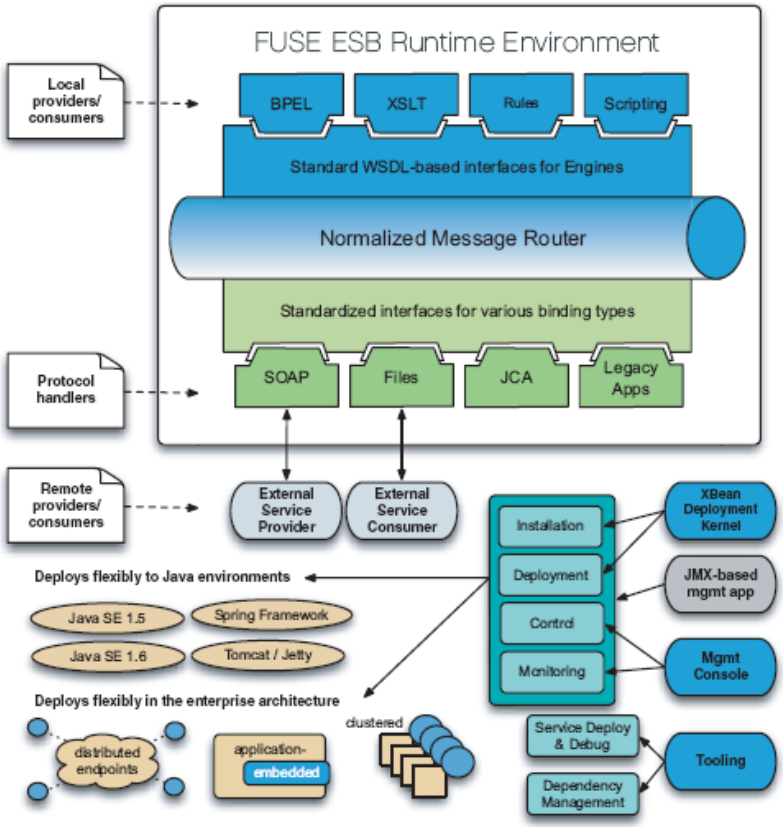
Oracle Database

Long-term data store

Design Goals:
- **Information-oriented service model**
- **Simplified data modeling**
- **Automated information management**

SWIM Service Container

- **Progress Software's FUSE product selected as the SWIM container in August 2008**
- **Open-Source Components**
  - **ESB: ServiceMix**
  - **Message Broker: ActiveMQ**
  - **Services Framework: CXF**
  - **Mediation Router: Camel**
- **Roles in TFDM**
  - **ESB: hosts ADMT modules, provides standard interfaces to external components**
  - **Message Broker: pub/sub infrastructure for inter-process communication**
  - **Services Framework: implements information sharing via Web Feature Service (WFS)**
  - **Mediation Router: assimilates external feeds such as FDIO, ASDE-X**

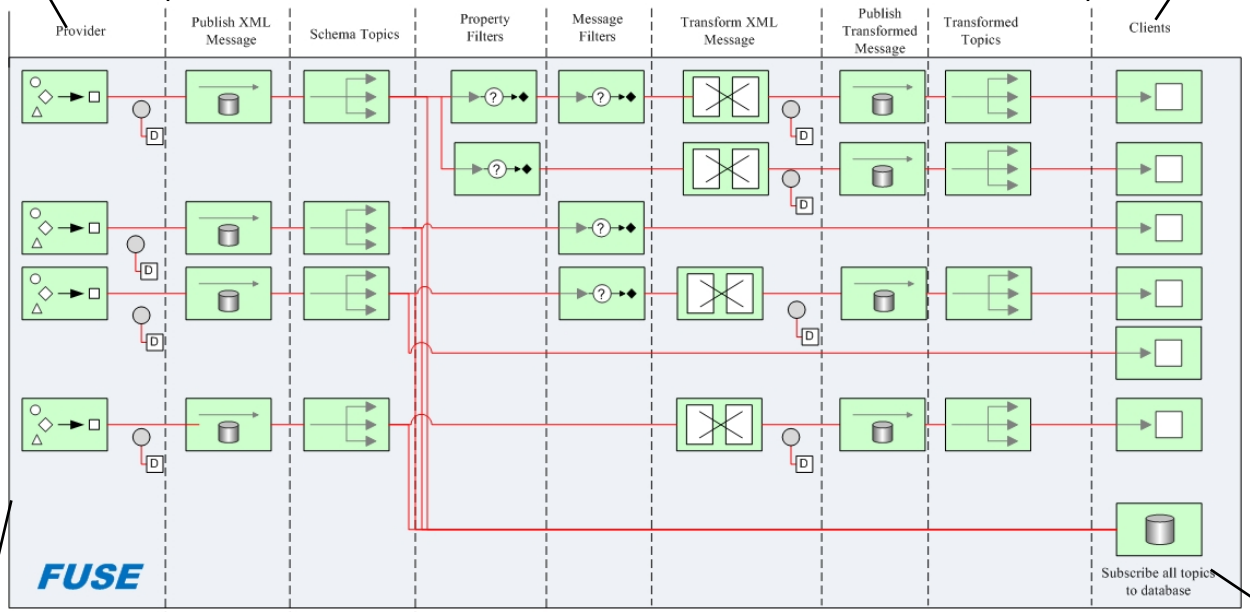


Initial TIMA Implementation in FUSE

Custom components for XML schema-based data normalization (e.g. ASDE-X)

"No-code' implementation of XML-Schema based "Information Bus" *

Custom client code for information processing



WFS-based publishers will be implemented as providers with no change to the core infrastructure

Oracle-based WFS implementation will subscribe to all sources for archiving

Implementation Goals:

- **Maximal use of existing FUSE components**
- **Provide standardized non-WFS solution for local information sharing**
- **Enable integration of WFS as an "add-on" capability**

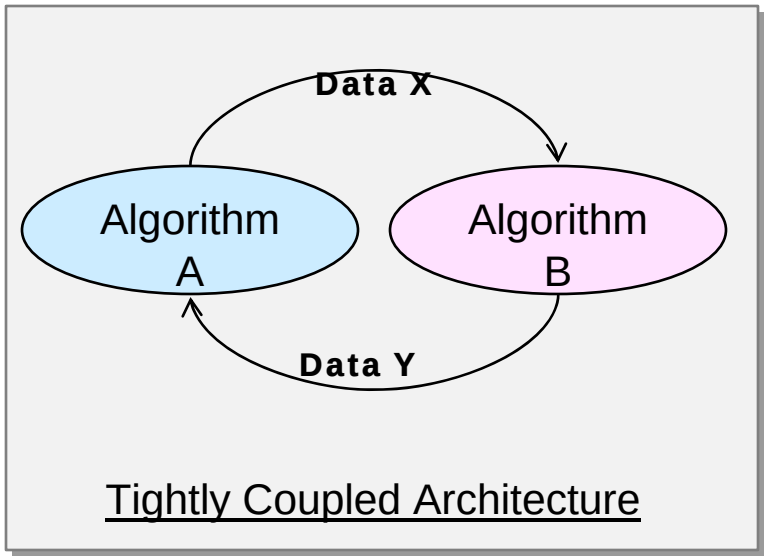
* requires configuration of Message Broker and Mediation Router Components

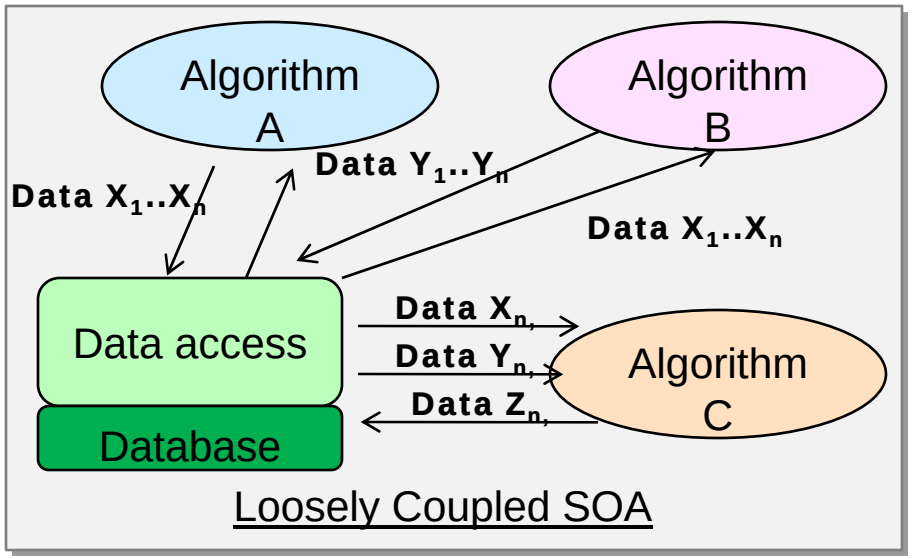
Technology Transfer

Tech Transfer Level	Advantages	Disadvantages
Concept	Integration into existing architecture not a problem	Process to get to implementation may be lengthy
Specification	Well-defined algorithms can be straightforward to implement	“Over-specification” can make implementation difficult for a given architecture
Implementation	Possibility that software modules can be reused	Redesign can be a slow process, as implicit specifications are reverse-engineered from the code

In all cases, we need a path for algorithm/technology enhancements

A/DMT Algorithm Architecture





- **Tightly coupled algorithms**
  - **Restrict data availability to a pre-determined set of peers**
  - **Tend to become brittle as enhancements are made**
- **Loosely coupled SOA approach**
  - **Provides open access model so existing information is easily tapped for new purposes**
  - **Preserves maintainability without sacrificing extensibility**

A/DMT technology insertion requires re-engineering of algorithms so that interactions are loosely coupled

Technology Transfer Stages

Capabilities
- Essential algorithms
- Module specifications
- As-built designs
- "Plug'n'Play components

Wide range of realizations, depending on SW Eng. Level of organization

Concept/Specification Implementation
- Design using loose-coupling
- Make information model explicit

Real-time Operations
- Prototype operational system

Algorithm Re-engineering
- Re-implement as loosely-coupled subsystem
- Re-design information model

Shadow-mode Operations
- Real-time operations
- Inputs are live data feeds
- OPEN-LOOP MODE ONLY – no feedback to TFDM system

Desktop Simulation
- Offline database-driven
- Modeled ATM entities
- Simple set-up and execution

Networked Simulation
- On-line, human-in-the-loop
- Real and emulated ATM entities, data sources, and protocols
- Distributed participants

Realistic simulation environments facilitate rapid deployment

Metron Aviation's JSE

Adacel's MaxSim

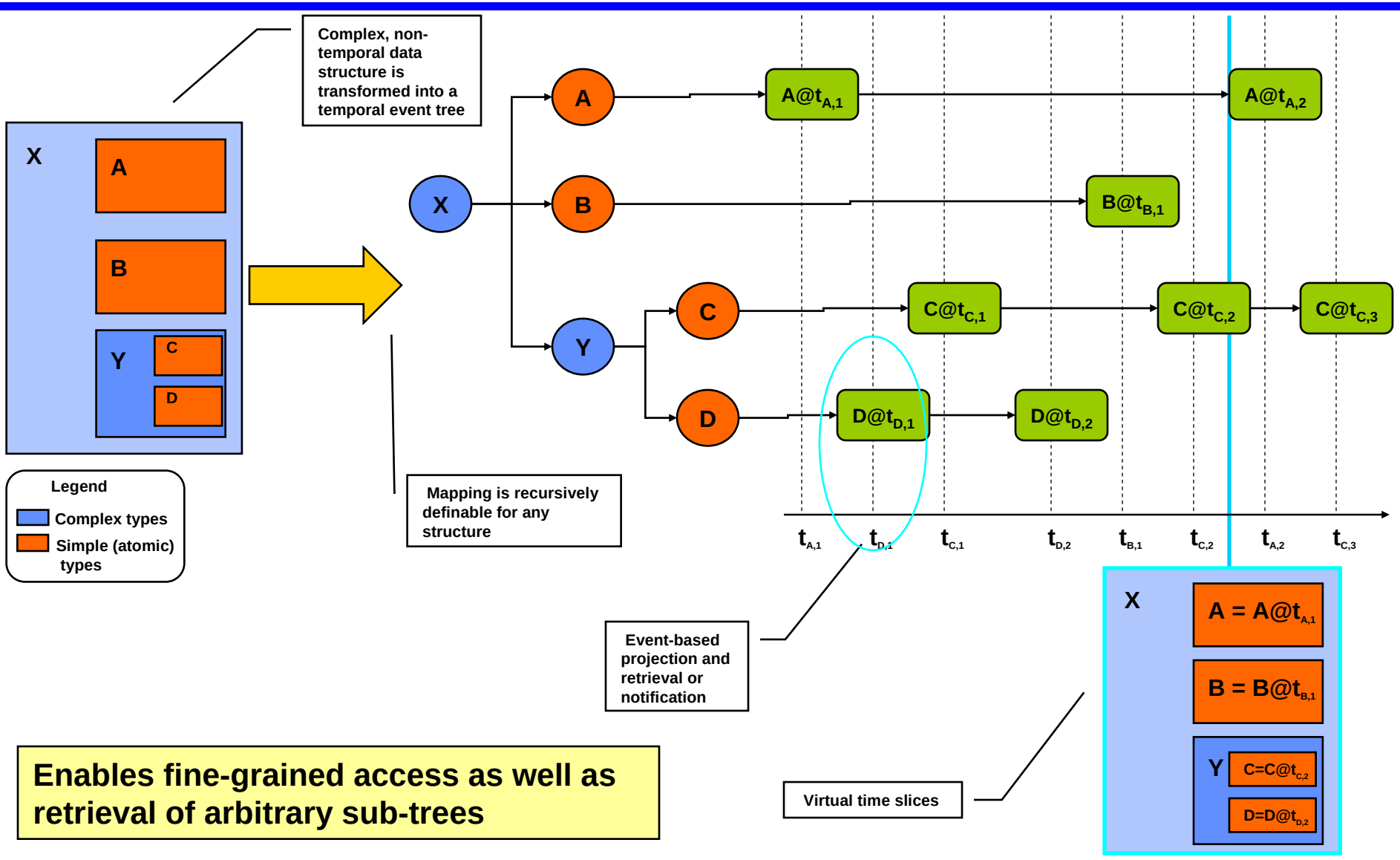
Temporal Model Design

- A uniform temporal reference for disparate data sets is critical to NextGen capability development and enhancement
  - Provides a basis for information alignment needed for real-time processing as well as off-line analysis, research, etc
  - Currently not a concern of the SWIM program

    Note that AIM program has a good initial attempt at capturing some temporality
- Temporal annotation structure can be produced in an "information agnostic" manner
  - Temporal "meta-schema" is applied to a "snapshot" information schema to produce a time-aware schema
  - Examples: transaction-time and valid-time
- "Event tree" temporal model provides both storage efficiency and access flexibility

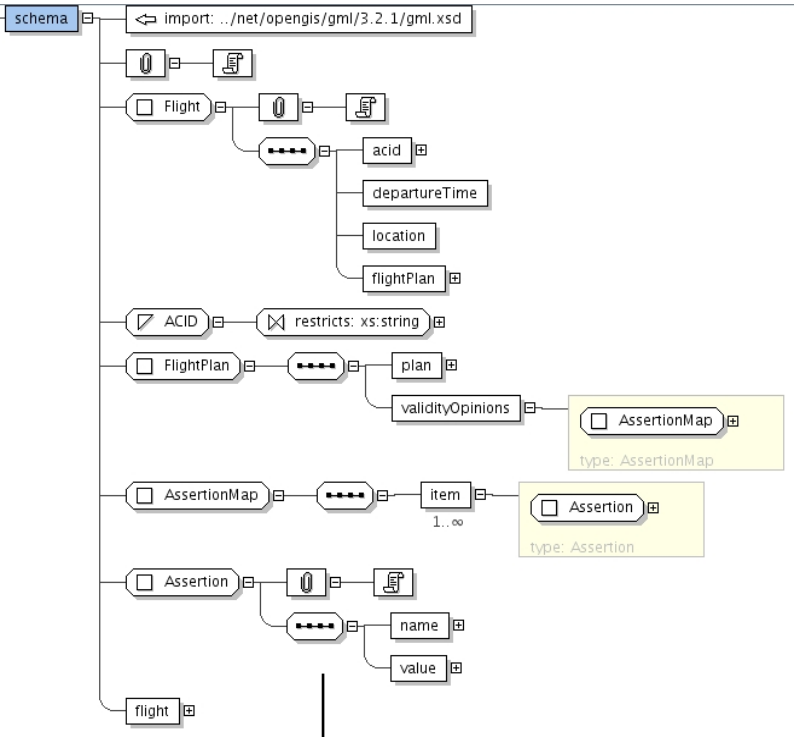
Goal: provide simple temporal modeling approach, while maintaining flexibility in information storage and retrieval

Mapping "snapshot" models to event trees



Enables fine-grained access as well as retrieval of arbitrary sub-trees

User-defined “Snapshot” Schema



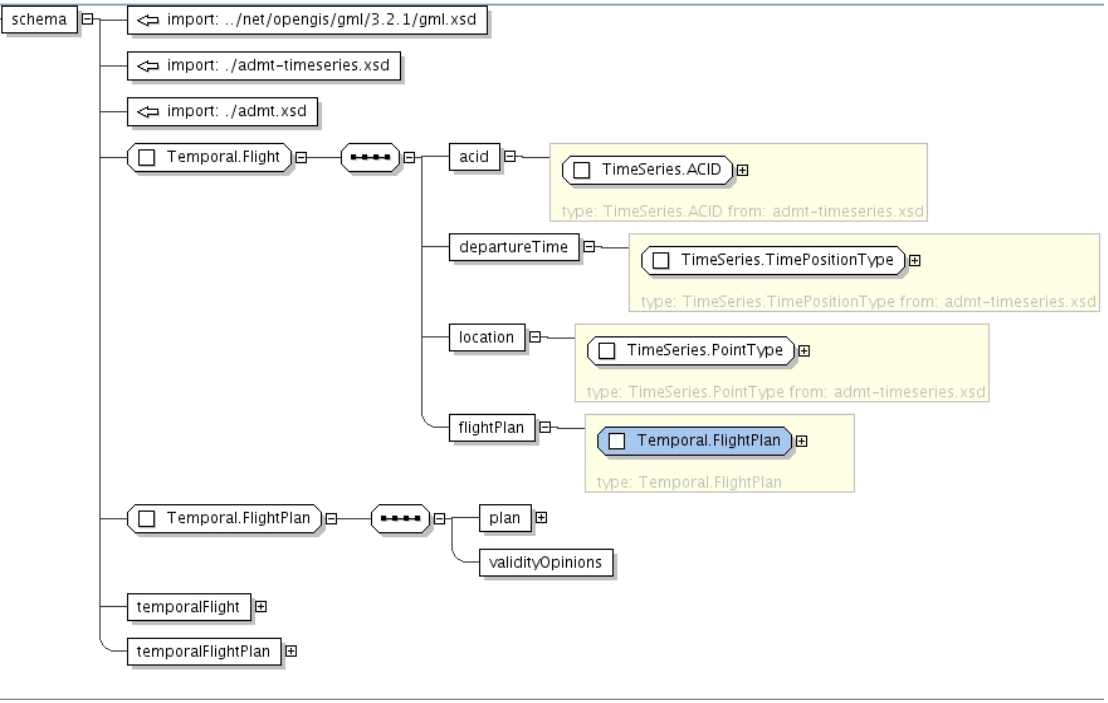
“Flight object” Schema definition

- **Schema has no “generic” temporal infrastructure**
  - **Ok to have data items of a temporal nature, such as *departureTime***
- **Any needed temporal infrastructure can be generated from this schema**

Schema instance (data for a flight)

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!-- instance of an admt:flight.  should  be easy to auto-generate a file like "flight-timeslice" from this -->
4  <admt:flight xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5     xsi:schemaLocation="http://wx.ll.mit.edu/admt admt.xsd" xmlns="http://wx.ll.mit.edu/admt"
6     xmlns:admt="http://wx.ll.mit.edu/admt" xmlns:gml="http://www.opengis.net/gml/3.2">
7     <acid>AA123</acid>
8     <departureTime>2008-10-03T00:00:06</departureTime>
9     <location gml:id="currentLocation">
10       <gml:coordinates>-102.1585083 35.3944664</gml:coordinates>
11    </location>
12    <flightPlan>
13       <plan>BOS.ORD</plan>
14       <validityOpinions>
15          <item>
16             <name>GroundControl</name>
17             <value>true</value>
18          </item>
19          <item>
20             <name>RampControl</name>
21             <value>true</value>
22          </item>
23       </validityOpinions>
24    </flightPlan>
25 </admt:flight>
26
```

A “Temporal” Flight Object




```
<?xml version="1.0" encoding="UTF-8"?>

<!-- instance of full temporal view of a flight object -->
<tm:temporalFlight xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://wx.ll.mit.edu/admt admt.xsd
                       http://wx.ll.mit.edu/admt/timeseries admt-timeseries.xsd
                       http://wx.ll.mit.edu/admt/temporal admt-temporal.xsd"
   xmlns:admt="http://wx.ll.mit.edu/admt"
   xmlns:ts="http://wx.ll.mit.edu/admt/timeseries"
   xmlns:tm="http://wx.ll.mit.edu/admt/temporal"
   xmlns:gml="http://www.opengis.net/gml/3.2">

   <tm:acid>
      <ts:item>
         <ts:time>2008-10-02T22:00:00</ts:time>
         <ts:value>AA123</ts:value>
      </ts:item>
   </tm:acid>
   <tm:departureTime>
      <ts:item>
         <ts:time>2008-10-02T22:00:00</ts:time>
         <ts:value>2008-10-03T00:00:00</ts:value>
      </ts:item>
      <ts:item>
         <ts:time>2008-10-02T23:45:00</ts:time>
         <ts:value>2008-10-03T00:10:00</ts:value>
      </ts:item>
   </tm:departureTime>

   <tm:location>
      <ts:item>
         <ts:time>2008-10-02T22:00:00</ts:time>
         <ts:value gml:id="location1">
            <gml:coordinates>-102.1585083 35.3944664</gml:coordinates>
         </ts:value>
      </ts:item>
   </tm:location>
   <tm:flightPlan>
      <tm:plan>
         <ts:item>
            <ts:time>2008-10-03T00:00:06</ts:time>
            <ts:value>BOS.ORD</ts:value>
         </ts:item>
      </tm:plan>
      <tm:validityOpinions>
         <ts:item>
            <ts:time>2008-10-02T22:00:00</ts:time>
            <ts:value>
               <admt:item>
                  <admt:name>GroundControl</admt:name>
                  <admt:value>true</admt:value>
               </admt:item>
            </ts:value>
         </ts:item>
         <ts:item>
            <ts:time>2008-10-03T00:00:06</ts:time>
            <ts:value>
               <admt:item>
                  <admt:name>GroundControl</admt:name>
                  <admt:value>true</admt:value>
               </admt:item>
               <admt:item>
                  <admt:name>RampControl</admt:name>
                  <admt:value>false</admt:value>
               </admt:item>
            </ts:value>
         </ts:item>
      </tm:validityOpinions>
   </tm:flightPlan>
</tm:temporalFlight>
```

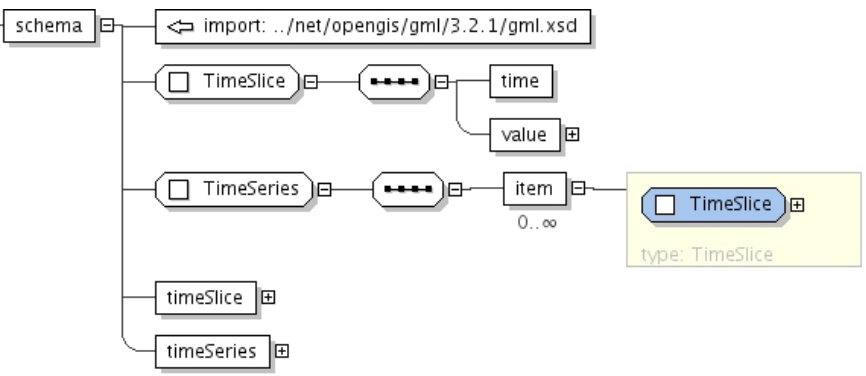
- **Flight Object Schema, revised to incorporate temporal infrastructure**
- **Provides incremental construction of full event tree**
- **Maps to time slices, event projections**
- **XML←→ DB mapping is straightforward, independent of original user data schema**

A Flight Object “Virtual Time Slice”

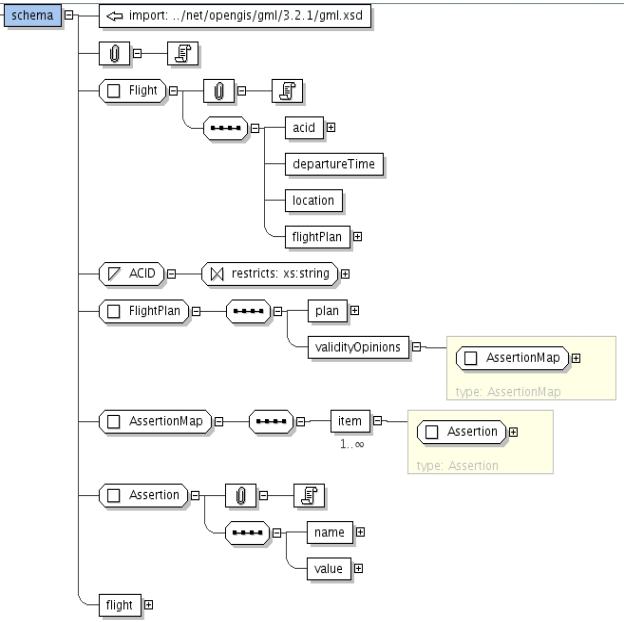
```xml
<?xml version="1.0" encoding="UTF-8"?>

<!-- "weakly-typed" timeslice: just an instance of the TimeSliceType in admt-timeseries.xsd -->
<ts:timeSlice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://wx.ll.mit.edu/admt admt.xsd
                        http://wx.ll.mit.edu/admt/timeseries admt-timeseries.xsd
                        http://wx.ll.mit.edu/admt/temporal admt-temporal.xsd"
   xmlns:admt="http://wx.ll.mit.edu/admt"
   xmlns:ts="http://wx.ll.mit.edu/admt/timeseries"
   xmlns:gml="http://www.opengis.net/gml/3.2">

   <ts:time>2008-10-02T23:00:00</ts:time>

   <ts:value>
      <admt:flight>
         <admt:acid>AA123</admt:acid>
         <admt:departureTime>2008-10-03T00:00:06</admt:departureTime>
         <admt:location gml:id="currentLocation">
            <gml:coordinates>-102.1585083 35.3944664</gml:coordinates>
         </admt:location>
         <admt:flightPlan>
            <admt:plan>BOS.ORD</admt:plan>
            <admt:validityOpinions>
               <admt:item>
                  <admt:name>GroundControl</admt:name>
                  <admt:value>true</admt:value>
               </admt:item>
               <admt:item>
                  <admt:name>RampControl</admt:name>
                  <admt:value>true</admt:value>
               </admt:item>
            </admt:validityOpinions>
         </admt:flightPlan>
      </admt:flight>

   </ts:value>

</ts:timeSlice>
```

- **The temporal event tree can be parsed to provide virtual time slices containing arbitrary projections of state at any time desired**
- **Output matches user-defined schema**

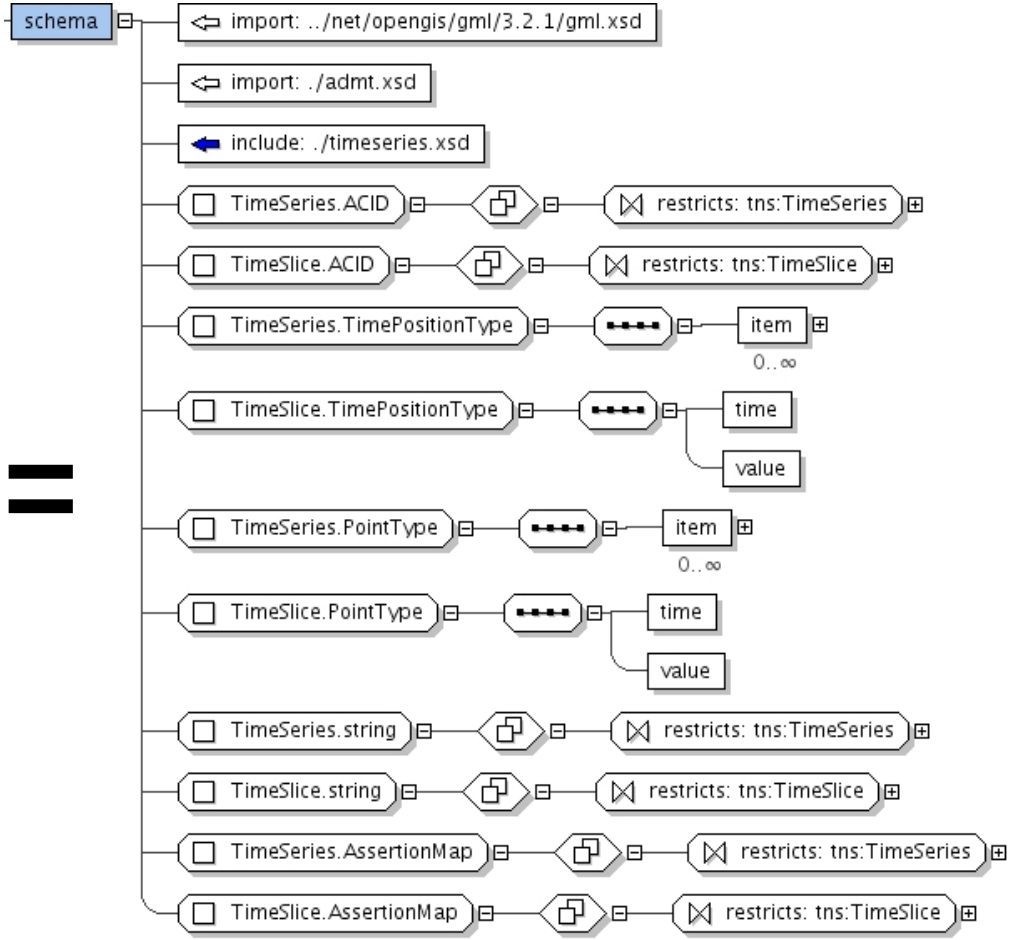
Generic Time Series infrastructure



+



=



Simple time-series schema serves as a template for schema based on user data types

Summary

- Preliminary TFDM Information Management Architecture has been developed
- Implementation underway utilizing standards-based approach
  - SWIM container (FUSE)
  - OGC's WFS
- Developing a process for moving A/DMT decision support algorithms from conception to TFDM implementation
  - Simulation-based approach is essential
- Temporal model being implemented as part of WFS extensions